

**FORSCHUNGSZENTRUM JÜLICH GmbH**  
**Zentralinstitut für Angewandte Mathematik**  
**D-52425 Jülich, Tel. (02461) 61-6402**

Interner Bericht

**On "Retarded Potentials" in  
High-Performance Scientific Computing**

*Friedel Hoßfeld*

KFA-ZAM-IB-9426

November 1994  
(Stand 22.11.94)

In G.-R. Hoffmann and N. Kreitz (eds.): "Coming of Age - The Use of Parallel Processing in Meteorology", Proceedings Sixth ECMWF Workshop on the Use of Parallel Processors in Meteorology, World Scientific Publishing Co., 1994, pp. 12-23.



# ON "RETARDED POTENTIALS" IN HIGH-PERFORMANCE SCIENTIFIC COMPUTING

FRIEDEL HOSSFELD  
*Central Institute for Applied Mathematics  
Research Centre Jülich (KFA)  
D-52425 Jülich, Germany*

## ABSTRACT

What particle accelerators mean to experimental physics, supercomputers mean to Computational Science and Engineering: *Supercomputers are the accelerators of theory!* Due to the high investments involved and challenged by the beneficial development of powerful workstation technology, the rationale of dinosaur vector-supercomputers is questioned while promises of near-by teraflops-sustaining massively parallel computers break in these days. Hence, and due to the inherent complexity of parallel computing, Computational Science and Engineering is suffering from severe retardations of efficient computer solutions; this is particularly painful because the analytical mathematical methods suffer from huge barriers to progress as well. Real-world supercomputing testifies, however, that the inherent variability of complex problems requires a non-uniform arsenal of powerful computing instruments suggesting to integrate complementary architectures into heterogeneous (meta)computers.

## 1. The Methodological Tripod of Science and Engineering

During the past decades, computer simulation has grown and established itself as the third category of scientific methodology; more comprehensively identified as *Computational Science and Engineering*, this innovative discipline fundamentally supplements and complements theory and experiment, as the two traditional categories of scientific investigation, in a qualitative and quantitative manner while integrating these into the *methodological tripod* of science and engineering. Being comparable rather with an experimental discipline, Computational Science and Engineering vastly extends the analytical techniques provided by theory and mathematics; today, in a sense, it is synonymous with investigating complex systems. Its main instrument is the supercomputer; its primary technique is computer simulation. Unsolved complex problems in the areas of climate research and weather forecast, chemical reactions and combustion, biochemistry, biology, environment and ecological as well as sociological systems, order-disorder phenomena in condensed-matter physics, astrophysics and cosmology, and, in particular, hydrodynamics have been identified as the *Grand Challenges*<sup>1</sup>.

The various strategic position papers<sup>2-4</sup> and government technology programs in the U.S., in Europe, and Japan claim that the timely provision of supercomputers to science and engineering and the ambitious development of innovative supercomputing hardware and software architectures as well as new algorithms and effective programming tools are an urgent research-strategic response to the grand challenges arising from these huge scientific and technological barriers<sup>5</sup>. As has been said, progress is a snail, and, probably, we cannot expect too fast successes in these areas despite the major technological and architectural steps which have been achieved in the design, engineering, and application of supercomputers just recently.

Scanning the history since the very birthday of Computational Science and Engineering, which may be dated back to 1946 when John von Neumann formulated the strategic program in his famous report on the necessity and future of digital computing together with H. H. Goldstine<sup>6</sup>, at that time complex systems were primarily involved with flow dynamics. Already John von Neumann summarized the situation of theoretical fluid dynamics by stating that "the advance of analysis is, at this moment, stagnant along the entire front of nonlinear problems", since "mathematicians had nearly exhausted analytic methods which apply mainly to linear differential equations and special geometries". He proposed that one should substitute numerical for analytical methods by supporting the development of digital computers and utilizing them, since digital devices could be made much faster and have more flexibility and more accuracy than 'analog computers' with which he also identified windtunnels.

He expected that really efficient high-speed digital computers will "break the stalemate created by the failure of the purely analytical approach to nonlinear problems" and suggested fluid mechanics as a source of problems through which a mathematical penetration into the area of nonlinear partial differential equations could be initiated. John von Neumann envisioned computer output as providing scientists with those heuristic hints needed in all parts of mathematics for genuine progress and to break the deadlock - "the present stalemate" - in fluid mechanics by giving clues to decisive mathematical ideas. In a sense, his arguments sound very young and familiar.

As far as fluid mechanics is concerned, in his John von Neumann Lecture at the SIAM National Meeting in 1981 yet Garrett Birkhoff came to the conclusion on the development of analytical fluid dynamics that it be unlikely that numerical fluid dynamics will become a truly mathematical science in the near future, although computers may soon rival windtunnels in their capabilities; both, however, will be essential for research<sup>7</sup>. While Birkhoff's analysis of the status of fluid dynamics, in principle, may be true yet today, there have been certainly dramatic expansions of numerical computing methods based on the rapid rise of supercomputers since then which yielded unforeseen insight into the subtleties of the nonlinear processes involved in flow-dynamic systems rather than producing just numbers<sup>8</sup>. Important milestones of Computational Science and Engineering have been passed also in many other fields included in the Grand Challenges and beyond.

This implies a host of innovative computer architecture designs and realizations as well as algorithms, numerical methods, programming models and tools in connection with language extensions and designs<sup>9</sup>.

The tripod of science and engineering, thus, has proved to provide scientific research and technology with the stable methodological basis and the instrumental laboratory to effectively approach the solutions of the complex problems which are crucial to the future of science, technology, and society. In order to reach the goals set by these grand challenges, it is essential to recognize that the scientific knowledge and the technical skills, which are available in the field of supercomputers and their applications and which will be further gained from scientific and technical engineering projects within universities and research institutions, will be a crucial factor for the industry in order to meet the requirements of international economic competition especially in the area of high-tech products. The present distribution of installed supercomputers over the industrial nations in America, Europe, and Japan may give more than a hint about a positive correlation between the availability of this innovative computing technology and their competitive-ness and economic power in the high-tech sectors.

Academia in the U.S. was pushing Computational Science and Engineering via a series of important strategic reports and initiatives - like the foundation of the NSF Supercomputer Centers and regional centers focussing on high-performance computing in interdisciplinary cooperation environments well suited to overcome the traditionally high barriers between faculties. Despite the remarkable investments in research centers and universities in building up supercomputing power and skills and also some sporadic efforts in the industry concerning supercomputing in Europe, it took until the 90s that the U.S. and European as well as national governments started non-military strategic support programs like HPCC, HPCN, and HPSC<sup>10-12</sup>. Their goals are also to enhance supercomputing as an innovative technology in science and engineering by stimulating the technology transfer from universities and research institutions into industry and by increasing the fraction of the technical community which gets the opportunity to develop the skills required to efficiently access the high-performance computing resources. This significant delay in the recognition and appreciation of high-performance computing as a revolutionary potential for science, technology, and, finally, industry may further retard their competitive capabilities to treat complex problems.

## **2. The Technological Pyramid of Scientific Computing**

Since the rise of vector-supercomputers in the late 70s they always have been expensive and costly facilities which, at least in scientific research and technical engineering, can be justified only for well-defined large research projects and, therefore, have been dedicated to a minority of scientists and engineers striving to expand the frontiers of their scientific and technological fields.

Due to the amount of investments involved, these super-computers as well as the classical mainframe computers have been challenged by the recent beneficial technological development of powerful workstations and by the emerging parallel computers relying on the growing power of the "killer-micros" as compute nodes. The rationale of the dinosaur vector-supercomputers has been questioned since then in the same manner as mainframes have been overcome by the client-server structures of open systems.

For the first time in computing history, we will be able to build a balanced pyramid of computing power in scientific and technical computation in which each element of the pyramid supports the others <sup>13</sup>. At the apex of the pyramid will be the highest level of compute power which can be realized by the computer architects and the industry with respect to efficient hardware and software targeting at the teraflops systems requested by the Grand Challenges.

This implies that, as a lower level of the pyramid and in order to develop the skills and the applications of future innovative computer architectures, universities and research institutions as well as industrial research divisions should be provided with mid-sized supercomputer systems. This level is required for the demanding science and engineering problems that do not need the very maximum of computing capacity, and for the computer science and computational mathematics community in order to take care of the architectural, operating systems, tools, and algorithmic issues which have built up primary barriers to progress especially in massively parallel supercomputing.

A third and, according to the structure of the pyramid, much broader level of scientific computing environments has to be supported by further major investments in order to provide science and research with the required infrastructure of powerful workstations as the effective workbench of scientists and engineers, in addition to the tremendous functionality of personal computers. It should no longer be a question that these facilities have to be networked campus-wide or corporate-wide with easy access to external communication services like Internet, which leads to the very basement of the pyramid - the network. Whereas local area and wide area network systems with medium speed and bandwidth have been built up almost everywhere in research institutions and universities, and high-speed communication with broadband functionality is promoted in the U.S. on a large scale for scientific as well as commercial applications and also in some European countries strong efforts are made to provide the scientific community with broadband communication services, e.g. in Great Britain with SuperJanet, other European countries are either still quite far from having access to broadband communications or just start to establish a few testbeds with innovative network systems, like with ATM in Germany while due to the high PTT tariffs many universities had even no chance so far to get interconnected to communication services with transmission rates of 2 megabits per second available since several years, e.g. due to the beneficial activities of the German Science Network (DFN).

There is broad consensus that the backlash in high-speed communications is a severe barrier to establishing a highly efficient nation-wide infrastructure which is capable to provide supercomputer capacity and functionality to the scientific community on a modern scale with transfer opportunities into the industry <sup>14</sup>. The lack of high-speed communications is certainly an important reason for the "retarded potential" of the high-performance computing technology in these countries. But other reasons for retardations in the promotion and progress of supercomputing seem to be latent.

### **3. The Decades of the Vector-Workhorses**

Strategically, for John von Neumann flow-dynamical phenomena have been the primary field where future efforts should have been invested to develop and establish the digital computer - as the "digital windtunnel" - and, thus, by utilizing numerical methods, activate the mathematical penetration of the whole area of partial differential equations. Since then and up to now partial differential equations have been dominating in the advancement of high-speed computers and in the exploitation of their potential.

The general solution methodology for such equations leads via discretization of space and time and via linearization into linear algebra and its numerical concepts and algorithms. The late response of computer architecture to these early challenges of PDEs have been the vectorcomputers optimizing vector-pipeline processing and creating the effective instruments of vectorization <sup>15</sup>.

Already in 1982, Cray Research made the significant step into multiprocessor vector-architectures and, hence, into parallel processing; simultaneously, the operating systems turned to open system technology supporting Unix functions and TCP/IP communications as the two innovative streams which emerged from the world of science and research as well and carry the development of computing on a broad wavefront - reaching from workstations via supercomputers to networking - into the future. It turned out that the Multitasking concept, in particular its recent way of Autotasking, in combination with powerful parallelizing compile systems and tools allow for efficient implementations of parallel algorithms thus providing the platform for gigaflops compute power also for large applications of the Grand Challenge class.

The exploration of the computing potential of the pipelining principle including programming and compiler techniques, tools, operating system functionality, and shared-memory organization and optimization resulted in the efficient arsenal of knowledge and experience about the strengths and the weaknesses of vectorcomputing. The highest class of vectorcomputers, e.g. the Cray, Fujitsu, and NEC systems, are still dominating the supercomputing production environments and the practice of Computational Science and Engineering <sup>16</sup>.

Certainly, vectorcomputers will further develop in functionality and performance towards the 100 gigaflops target by exploiting the architectural and technological potential and expanding the "weak" parallelism well beyond the presently 16 processors. e.g. of the CRAY C-90.

Even today, the sustained performance of these systems, e.g. the CRAY C-90 or NEC SX-3, turns out to be still far ahead of the sustained performance of massively parallel systems for a vast majority of essential algorithms as well as large applications. Therefore, despite the relative progress of massively parallel computers, the very workhorses of Computational Science and Engineering today are still vectorcomputers.

#### **4. The Promises of Massive Parallelism**

The rise of powerful workstations and the possibility to interconnect them quite easily to clusters, via Ethernet and even more efficient communications, has created an amount of compute capacity in many places which has not been experienced ever before in the times of mainframes and "stupid" terminals attached to them. Suddenly, big applications and simulation runs could be performed on these individual workstations, and the clusters could even be utilized as a kind of parallel computer if applying valuable software concepts and systems like PVM or others. The price-to-performance arguments are still very strong, at least when "farming" of the applications is hiding the otherwise efficiency-killing latency times.

From these experiences aggressive attacks have been generated against conventional vector-supercomputing and against massively parallel systems as well, in particular against those which were and still are suffering from low compute power of the processor nodes and bad software and stability. It will take quite some time to recover from these irritations which have slowed down the engagement in promoting genuine parallel computer architectures well beyond workstation clusters.

Fortunately, forthcoming parallel computer structures like the IBM SP-2 while utilizing workstation technology overcome the weakness of workstation clusters for genuinely parallel algorithms; they provide at least part of a solution to the cluster deficiencies by utilizing high-speed switches. There is no doubt that today's RISC-based workstations are definitely killing most of the rationale of the classical general-purpose mainframe systems; their proprietary operating systems cause permanently difficulties to keep up with the progress into open systems technology which offers the attractive elements of a new computing culture within scientific and research environments and which will soon penetrate commercial data processing as well. Workstations and workstation clusters, on the other hand, provide the excellent capacity to free the higher-class supercomputers from the increasing number of "small" supercomputer applications by off-loading, thus reserving them for the really large applications of the Grand Challenge category which can justify the high expenditures of the numerical "windtunnels" or "accelerators".



But workstations, however powerful they are or will become, cannot replace the potential of parallel computers which are basically built upon this technology of powerful microprocessor chips by tying them together via sophisticated broadbanded interconnection networks in order to support massive parallelism. Massively parallel computers are therefore undoubtedly considered as the - only - remedy to the needs of the demanding applications in the Grand Challenge category and maybe yet unrecognized applications which might emerge, for instance, from the expanding multimedia field already today.

The different european and national research initiatives almost exclusively target at the advancement of massively parallel computer architectures and the technology transfer of parallel computing into industrial applications.

Unfortunately, in the early 90s the manufacturers of massively parallel systems promised that they would be capable to develop and deliver parallel supercomputers in 1995 which be able to reach the magical "3 T's" (i.e. 1 Teraflops in execution rate, 1 Terabyte in main memory, and 1 Terabyte/s interconnection bandwidth), thus indicating a revolutionary, rather than evolutionary, step of almost three orders of magnitude beyond the current state-of-the-art supercomputer performance.

During recent years, nearly thirty companies were offering massively parallel systems and others were planning to enter the market with new products, although many experts predicted that the market will not be able to sustain this many vendors<sup>17</sup>. In the meanwhile, there has not only started the expected shake-out in the respective computer industry questioning the health and the future potential of this industry in total; the fundamental reasons for the dramatic survival battle in the supercomputer industry are also giving severe damage to the users of parallel computing facilities.

Their investments into this massively parallel computing strategy may be definitely lost and the establishment of a new hardware and software platform will require new investments concerning finances and manpower as well as psychological recovery from the frustration caused by the unfulfilled soap-bubble promises of several, if not all, manufacturers in this field. This certainly will result in an even more painful retardation of the fruitful exploitation of the potential of parallel computing.

Just recently, John Gurd<sup>18</sup> has analyzed the critical situation of parallel computing and the possible negative impacts on the future perspectives and the progress of this scientific discipline but also on the support which will be expected and requested from the politicians. Although he states that "the history of computing is littered with failed long-term predictions", he is right in claiming honest answers from the supercomputing arena to some burning questions on the seriousness of predictions concerning the reachability of the goals set in particular in the context of those national research initiatives. One might be tempted to recommend to everybody in the field to obligatorily read Gurd's paper before going on in his work!

The key issue in massively parallel computing is scalability. Parallelizing "dusty" decks from industry is certainly an important task to do in order to increase the acceptance of parallel computing in commercial environments. However, one cannot expect terrific performance gains in many of these programs from porting such ordinary sequential, in many cases also organically grown, codes to parallel systems. Therefore, scalability often breaks down when the number of parallel nodes is increased beyond sixty-four, sixteen or even eight which cannot be said to be massively parallel in the very sense. Even the benchmark results on naked algorithmic kernels stress the limiting factor of scalability.

There is a big discrepancy between the peak rates of massively parallel systems and the sustained performance which can be reached with algorithmic kernels and, even more significantly, with real application programs and software packages <sup>19</sup>. With kernels, the state of the art of massively parallel computers delivers, together with a pretty large variance in the performance data depending on the definite architecture of the system and the algorithm as well, in the average around 10% of the peak rate as sustained performance. This is certainly a disappointing fact. But taking this average performance into account, the price-to-performance ratio of massively parallel computers is losing its attractiveness if compared with vectorcomputers, too, which has been one of the striking arguments in favour of massively parallel systems.

Since scalability is critical as well, only few applications are capable to exploit massive parallelism up to a scale where vectorcomputers become definitely inferior with respect to sustained performance. It has become clear that the trend is coming back to more powerful nodes in parallel systems, rather than promoting transputer-level node performance.

Since, so far, the microprocessor chips have been developed with a different market goal in mind, it is extremely difficult to exploit the performance hidden in the hardware design of these processors via high-level programming languages and compiler techniques <sup>20</sup>; very often this leads to a loss by a factor of at least five to ten referred to peak performance of the node <sup>21-22</sup>. It cannot be accepted as a reasonable software-technological approach to switch back to the very old times of assembler programming to reach reasonable performance levels. Convergence of hardware and compiler design together with the development of valuable programming tools must become the future development strategy.

Another important issue is programming models. While Message Passing is widely and effectively used on distributed memory systems as the only efficiently implemented programming paradigm at present, one can hardly imagine that this programming model will carry all future efforts to introduce massively parallel computing as the overwhelming technology; especially large non-scientific applications will certainly suffer from this obstacle of explicit programming the data communication in message-passing style. Up to now, programming and software technology not only relied on sequential machines, but also on the shared-memory organizational concept.

The dominant language in scientific programming has been Fortran with all its strengths and weaknesses. While criticizing this language, the emerging new standard FORTRAN 90 provides some interesting features with respect to software-technological aspects; unfortunately, new compilers have had significant delays to be of benefit already. On this language basis, the forthcoming High Performance Fortran (HPF) seems to be an at least temporary platform to implement parallel applications. Although first implementations of the small HPF subset are just arriving, urgent extensions towards more functionality are scheduled for HPF-2 which might not appear within the next two years.

Despite the failure of the first commercially available massively parallel computer system which supported the programming paradigm of the Shared Virtual Memory (SVM), the efforts to explore this programming model should not be reduced or turned totally down. From a user's point of view, but also, maybe, from the language point of view this SVM paradigm<sup>23</sup> seems to carry enough potential to overcome fundamental deficiencies which can be experienced with the Message-Passing paradigm, because SVM hides the physical local memories from the programmer and provides a virtual address space organized in pages which demand-driven move across the parallel processors. Each local memory acts as a large cache. The advantage of this paradigm may be especially comfortable for the programmer if dealing with irregular data which are inherent in many scientific and engineering applications like, for instance, in the finite-element methods to treat partial differential equations<sup>24</sup>. Since in many cases data access and communication patterns are unknown prior to the parallel execution, data with efficient domain decomposition cannot be realized in advance as is required by the Message-Passing model. However, much research and development work has to be completed to achieve SVM implementations with tolerable overhead; in addition, this needs powerful tools to monitor the progress in the parallelization process and strong support on the hardware level which cannot be seen to be available in the near future due to the present lack of interest in this SVM paradigm on the manufacturers' side.

In any case, together with genuinely parallel algorithms, powerful and user-friendly programming tools as well as performance-monitoring capabilities are a key issue<sup>25</sup>.

#### **4. The Remedy of Heterogeneous (Meta)Computing**

The experiences with the different architectures available for the supercomputer applications in Computational Science and Engineering with their strengths and weaknesses, the technological obstacles for major performance steps in vector-computing, the large variance in the performance data for algorithms on different parallel machines, and the very low average sustained performance in massively parallel processing relative to the peak rate, the present or even fundamental limitations to the scalability of systems and applications to reach and exploit massive parallelism, quite naturally lead to the concept of heterogeneous computer systems which requires the coexistence and cooperation of the different computer architectures.

On heterogeneous systems <sup>26-28</sup>, the computational work of - parallel - programs can be split across different computers in order to achieve in total the fastest possible execution where the individual portions of the work are sent to those computer systems which have been proved to be best for the specific characteristics of the work. As an example, in a climate-modeling application the incoming earth-observation satellite data might be filtered on a more traditional (mainframe) system with strong I/O and file handling capabilities, then pipelined to a number-crunching vector-computer, passed to a highly parallel computer to convert the data into image renderings, and end up displayed as a series of animation frames on a high-performance workstation.

This approach can generate results much faster than would be possible on any one system. It also might simplify the programming effort, since program components can be developed using diverse software environments which usually are not available on any single machine. Heterogeneous computing is an attractive concept because it takes into account that the individual parallel machines, and vectorcomputers as well, spend much of their time on tasks for which they are unsuited; these effects lead to the experienced break-downs in sustained performance and also to scalability problems. On the other hand, it is well known that a user generally invests tremendous efforts in order to extract even that small level of sustained performance out of an innovative computer system for his specific application well knowing that the application principally implies a spectrum of heterogeneous requirements which cannot be efficiently satisfied by the single target system he has been focussing on just because it is available to him in his specific computing environment.

Since the performance of the known supercomputer architectures is a function of the inherent structures of the computations to be executed and the data communications involved, it is necessary to discriminate among types of code, algorithms, data, and communications in order to optimize the mapping of tasks onto computer structures. Researchers in the field of innovative computing believe that there will be no single all-encompassing architecture which will be capable to satisfy heterogeneous requirements with equally optimal performance.

The goal of heterogeneous computing is the efficiency of computation and thereby the effectiveness and cost-effectiveness of both computers and programmers. The price to pay are again grand challenges to respond by hardware and software as well as network designers; hence, certainly time delays will again retard the potentiality of getting heterogeneous computing into efficient use, not to speak of the even more cumbersome administrative and accounting problems involved in this cross-regional and inter-institutional computing concept. It becomes clear that high-speed networking is the fundamental technical requirement of heterogeneous computing on the way to meta-computing, which brings us back to broadband data communications as the very basis of the "technological pyramid of scientific computing" as discussed earlier in this paper.

## 5. References

1. Special Double Issue: *Grand Challenges to Computational Science*, Future Generation Computer Systems **5** (1989), No. 2&3.
2. Committee on Physical, Mathematical, and Engineering Sciences, Federal Coordinating Council for Science, Engineering, and Technology, *Grand Challenges 1993: High Performance Computing and Communications*, The FY 1993 U.S. Research and Development Program, Office of Science and Technology Policy, Washington, 1992.
3. Board on Mathematical Sciences of the National Research Council (USA), *The David II Report: Renewing U.S. Mathematics - A Plan for the 1990s*, in: Notices of the American Mathematical Society, May/June 1990, 542-546; September 1990, 813-837; October 1990, 984-1004.
4. Commission of the European Communities, Report of the EEC Working Group on High-Performance Computing (Chairman: C. Rubbia), February 1991.
5. U. Trottenberg et al., *Situation und Erfordernisse des wissenschaftlichen Höchstleistungsrechnens in Deutschland - Memorandum zur Initiative High Performance Scientific Computing (HPSC)*, Februar 1992; published in: Informatik-Spektrum **15** (1992), H. 4, 218.
6. J. von Neumann and H. H. Goldstine, in: Collected Works Vols. V/VI, Pergamon Press, 1983.
7. G. Birkhoff, *Numerical Fluid Dynamics*, SIAM Review **25** (1983), 1.
8. K. Kuwahara, *Flow Simulation on Supercomputers and its Visualization*, Intern. J. High Speed Computing **4** (1992), No. 1, 49.
9. K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, 1993.
10. The Congress of the United States, Congressional Budget Office, *Promoting High-Performance Computing and Communications*, Washington, June 1993.
11. High-Performance Computing Applications Requirements Group, *High-Performance Computing and Networking*, Report, European Union, April 1994. - High-Performance Networking Requirements Group, Report, European Union, April 1994.
12. Bundesministerium für Forschung und Technologie, *Initiative zur Förderung des parallelen Höchstleistungsrechnens in Wissenschaft und Wirtschaft*, BMFT, Bonn, Juni 1993.
13. NSF Blue Ribbon Panel on High Performance Computing, *From Desktop To Teraflop: Exploiting the U.S. Lead in High Performance Computing*, Report, NSF-CISE, 19 October 1993.
14. R. Bayer, *Plädoyer für eine Nationale Informations-Infrastruktur*, Informatik-Spektrum **17** (1994), H. 5, 302.
15. F. Hossfeld, *Vector-Supercomputers*, Parallel Computing **7** (1988), 373.
16. J. J. Dongarra, H.-W. Meuer and E. Strohmaier (eds.), *TOP500 Report 1993; TOP500 Report 1994*, University of Mannheim, Germany, May/November 1994.

17. The Superperformance Computing Service, Palo Alto Management Group, Inc., *Massively Parallel Processing: Computing for the 90s*, SPCS Report 25, Second Edition, Mountain View, California, June 1994.
18. J. R. Gurd, *Supercomputing: Big Bang or Steady State Growth?*, Keynote Address, 7th International Conference on Supercomputing, ICS'93, Tokyo, in: ACM Computer Architecture News **22** (1994), No. 3, 3.
19. D. H. Bailey et al., *NAS Parallel Benchmark Results 3-94; NAS Parallel Benchmarks Update 9-94*, RNR Technical Report RNR-94-006, NASA Ames Research Center, Moffett Field, California, March and September 1994.
20. H. Zima and B. Chapman, *Supercompilers for Parallel and Vector Computers*, ACM Press, 1990.
21. A. Mlynski-Wiese, *Leistungsuntersuchung des iPSC/860 RISC-Knoten-Prozessors: Architekturanalyse und Programmoptimierung*, Zentralinstitut für Angewandte Mathematik, Berichte des Forschungszentrums Jülich Jül-2766, Mai 1993.
22. A. Krumme, *Performance-Analyse des DEC Alpha-Chip: Moderne Compiler-Techniken zur Programmoptimierung*, Zentralinstitut für Angewandte Mathematik, Berichte des Forschungszentrums Jülich Jül-2912, Mai 1994.
23. K. Li, *Shared Virtual Memory on Loosely Coupled Multiprocessors*, PhD Thesis, Yale University, September 1986.
24. R. Berrendorf, M. Gerndt, Z. Lahjomri and Thierry Priol, *A Comparison of Shared Virtual Memory and Message Passing Programming Techniques Based on a Finite Element Application*, in: B. Buchberger and J. Volkert (eds.), *Parallel Processing: CONPAR 94 - VAPPVI*, Lecture Notes in Computer Science Vol. 854, Springer Verlag, 461-472.
25. W. E. Nagel and A. Arnold, *Performance Visualization of Parallel Programs: The PARvis Environment*, Proceedings 1994 Intel Supercomputer Users Group (ISUG) Conference, 24-31; Technical Report CCSF-47, Concurrent Supercomputing Consortium, Pasadena, California, May 1994.
26. Ch. Pancake, *Area Editor's Message: High-Performance Computing - Lighting the Way*, IEEE Computer **26** (1993), No. 6, 6.
27. R. F. Freund and H. J. Siegel, *Guest Editors' Introduction: Heterogeneous Processing*, IEEE Computer **26** (1993), No. 6, 13.
28. A. A. Khokhar et al., *Heterogeneous Computing: Challenges and Opportunities*, IEEE Computer **26** (1993), No. 6, 18.